NAME: _____

COMPSCI 611
Advanced Algorithms
Final Exam Fall 2017: Draft Solutions

A. McGregor 20 December 2017

DIRECTIONS:

- Do not turn over the page until you are told to do so.
- This is a *closed book exam.* No communicating with other students, or looking at notes, or using electronic devices. You may ask the professor to clarify the meaning of a question but do so in a way that causes minimal disruption.
- If you finish early, you may leave early but do so as quietly as possible. The exam script should be given to the professor.
- There are five questions. Some questions may be easier than others. Don't spend too long on a problem if you're stuck since you may find that there are other easier questions.
- The front and back of the pages can be used for solutions. There is also a blank page at the end that can be used. If you are using these pages, clearly indicate which question you're answering.
- Good luck!

1	/14
2	/10
3	/10
4	/10
5	/8+2
Total	/52+2

Question 1 (True or False): For each of the following statements, indicate whether the statement is true or false by circling the appropriate option. It is not necessary to give justification.

1. In class, we proved that $P \neq NP$.

   TRUE FALSE

2. It is possible to find the minimum spanning tree of a graph in polynomial time.

   TRUE FALSE

3. The maximum value of $3 - x_1 - x_2 + x_3$ subject to the constraints $x_1 + x_2 + x_3 \leq 1$, $x_1 \geq 0, x_2 \geq 0$, and $x_3 \geq 0$ is achieved when $x_1 = x_2 = x_3 = 0$.

   TRUE FALSE

4. If $T(n) = 2T(n-1) + 1$ and $T(1) = 1$ then $T(n) = 2^n - 1$.

   TRUE FALSE

5. For any random variable X where $\mathbb{E}[X] \geq \sqrt{\mathbb{V}[X]}$ then $\mathbb{P}[X \geq 3\mathbb{E}[X]] \leq 1/4$.

   TRUE FALSE

   Answer: Assuming X doesn't always equal 0, $\mathbb{P}[X \geq 3\mathbb{E}[X]] \leq \mathbb{P}[|X - \mathbb{E}[X]| \geq 2\mathbb{E}[X]] \leq \mathbb{V}[X]/(4\mathbb{E}[X]^2) \leq 1/4$. If X always equals 0 then the statement is false. So we marked your answer correct either way.

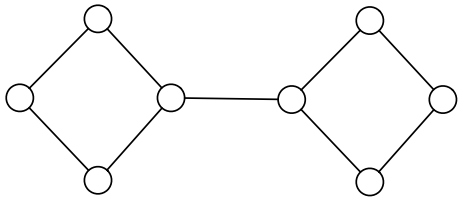
6. A subset system satisfies the exchange property iff it satisfies the cardinality property.

   TRUE FALSE

7. If a randomized algorithm is expected to terminate in 8 minutes then the algorithm will terminate in less than 24 minutes with probability at least 2/3.

   TRUE FALSE

Question 2: Consider the following graph for the first four parts of this question:



1. What are the following values for the above graph:

   size of the min cut $= 1$ size of the max cut $= 9$

2. If you contract a random edge, what's the probability the min-cut size stays the same?
   Answer: $8/9$

3. What's the minimum number of colors required such that it is possible to color each vertex with one of these colors such that the endpoints of every edge get different colors?
   Answer: 2

4. Suppose every vertex is given a color chosen uniformly at random from {red, blue, green}. What's the expected number of edges whose endpoints are colored different colors?
   Answer: $9 \times (2/3) = 6$

5. We now consider an arbitrary graph H with 150 edges. If it is possible to color each vertex with one of three different colors such that the endpoints of every edge receive different colors, what is the smallest the size of the max cut can be?
   Answer: 100. Since if R, B, G are the sets of vertices colored red, blue, and green then

   $$\text{size of cut } (R, V \setminus R) + \text{size of cut } (B, V \setminus B) + \text{size of cut } (G, V \setminus G) = 2 \times 150$$

   and hence one of these cuts must have size at least $2 \times 150/3 = 100$.

Question 3: Suppose a student is moving to Amherst to start graduate school. They have n items they need to pack numbered $1, 2, \ldots, n$ and the ith item has weight $w_i > 0$. Let $W = \sum_{i=1}^{n} w_i$. The weights are not sorted in any particular order. The student needs to put the items in packing boxes and each packing box can carry at most C kilograms and $C \geq w_i$ for all i. It is in the student's interest to use as few packing boxes as possible.

1. For this part only, suppose $C = 5, n = 4$, and the weights are $w_1 = 3, w_2 = 3, w_3 = 2$ and $w_4 = 4$. What's the smallest number of boxes that can used to pack all items?

   Answer: 3 boxes is the smallest possible, e.g., the first box gets item 1, the second box gets items 2 and 3, and the last box gets item 4.

2. Explain why the student needs to use at least $b^* = W/C$ packing boxes.

   Answer: Suppose that it was possible to use $b < W/C$ packing boxes. Then these boxes would only be able to carry weight $bC < W$ which means we haven't packed all items.

3. We say a box is *non-empty* if it has at least one item in it. The student uses the following algorithm to pack the items one at a time: She adds the next item to a non-empty box if doing so does not exceed weight C in that box; if there are no such boxes, see adds the next item to a box that is currently empty. Indicate whether each of the statements are always true or may be false.

All non-empty boxes have weight at least $C/3$.	TRUE	FALSE
At most one non-empty box has weight less than $C/2$.	TRUE	FALSE
The number of non-empty boxes is at most $3n/4$.	TRUE	FALSE
There are no boxes with weight strictly larger than C.	TRUE	FALSE

4. Suppose the above algorithm ends up creating b non-empty boxes. Prove the best upper bound on b/b^* you can. To get full marks your bound should not depend on W or C but you may assume W/C is an integer. Show your working and you may use your answers to the above TRUE/FALSE questions without justification assuming they are correct.

   Answer: At most one non-empty box has weight less than $C/2$. This implies that $(b-1)$ boxes each weight at least $C/2$. Hence, $(b-1)C/2 < W$ where the inequality is strictly because the other box has non-zero weight. Rearranging this gives $b < 1 + 2W/C$. Since W/C is an integer we know $b \leq 2W/C$. Hence, $b/b^* \leq (2W/C)/b^* = 2$

Question 4: A humanities professor enlists you to write a plagiarism detector based on common subsequences. We say two sequences have a common subsequence of length k if it is possible to delete all but k characters from each sequence, such that the resulting sequences are equal. For example, "*shewrotethis*" and "*hecopiedbits*" have a common subsequence of length 6, i.e., "*heoeis*". The idea is to treat essays as sequences and alert the professor to possible plagiarism if the sequences have a long common subsequence.

1. Complete the following algorithm for finding the length of the longest common subsequence (LCS) of two sequences $x_1 \dots x_n$ and $y_1 \dots y_n$ of length n. Let $D[i, j]$ be the length of the LCS of $x_1 x_2 \dots x_i$ and $y_1 y_2 \dots y_j$.

   initialize $D[0, \ell] \leftarrow 0$ and $D[\ell, 0] \leftarrow 0$ for all $\ell = 0$ to n
   for $i = 1$ to n **do**
      for $j = 1$ to n **do**
         if $x_i = y_j$ **then**
            $D[i, j] \leftarrow 1 + D[i-1, j-1]$
         else
            $D[i, j] \leftarrow \max(D[i-1, j], D[i, j-1])$
         end if
      end for
   end for
   return $D[n, n]$.

2. What's the running time of the above algorithm?
   Answer: $O(n^2)$.

We need to fix a threshold T such that if the LCS has length strictly bigger than T, we declare plagiarism to be suspected. If T is too low, there are many false alarms. In particular, we want it to be large enough such that two random sequences don't have a common subsequence of length T. Let x and y be random sequences of length n, i.e., each character is equally likely to be any of the 26 letters in the alphabet.

1. What's the probability the first T characters of x equals the first T characters of y?
   Answer: $1/26^T$.

2. Prove an upper bound on the probability that x and y have a common subsequence of length $T = en/\sqrt{13}$. **Hint:** You can use the bound $\binom{n}{T} \leq (ne/T)^T$ without proof.
   Answer: There are $\binom{n}{T}$ ways to pick a subsequence of a length n string. Hence, there are $\binom{n}{T}^2$ ways to pick a pair of subsequences, one from x and one from y. Each of them has a $1/26^T$ possibility of being equal. Hence by the union bound, the probability there exists a common subsequence of length T is at most

$$\binom{n}{T}^2 \times 1/26^T \leq (ne/T)^{2T} \times 1/26^T = 13^T/26^T = 1/2^T \ .$$

Question 5: We now consider problems related to VertexCover. Let $G = (V, E)$ be a graph with n vertices where every vertex has at least one neighbor.

1. Let $v(G)$ be the size of the minimum vertex cover. Indicate whether each of the statements are always true or may be false.

$v(G)$ is at least the size of a maximum matching.	TRUE	FALSE
The size of a maximum independent set is exactly $n - v(G)$	TRUE	FALSE

2. How can you define n sets $S_1, \ldots, S_n$ based on G such that for any integer k, G has a vertex cover of size k iff there exists k sets amongst $S_1, \ldots, S_n$ whose union equals E? This would show VertexCover $\leq_P$ SetCover.

$S_i =$ the set of edges that have the ith vertex as an endpoint

3. Prove there is a poly-time reduction from SetCover to VertexCover using the facts: a) 3-SAT is NP-complete, b) 3-SAT $\leq_P$ VertexCover, and c) SetCover $\in$ NP.

   Answer: SetCover $\leq_P$ 3-SAT since 3-SAT is NP-complete and SetCover $\in$ NP. Then, SetCover $\leq_P$ 3-SAT and 3-SAT $\leq_P$ VertexCover together imply

   SetCover $\leq_P$ VertexCover .

4. We say a set of edges *covers* a vertex v if v is an endpoint of at least one edge in the set. We say E' is an *edge cover* of G if it covers every vertex in the graph. Let m^* be the size of the maximum matching in G. Design a poly-time algorithm that finds an edge cover of size exactly $n - m^*$. You may assume the existence of a poly-time algorithm that finds a maximum matching in an arbitrary graph. Argue the correctness of the algorithm.

   Answer: Construct a matching of size m^*. There are $n - 2m^*$ vertices that aren't covered by these edges. For each vertex that isn't covered add an edge that has this vertex as an endpoint. This adds exactly $n - 2m^*$ edges and at this point we have an edge cover of size $m^* + (n - 2m^*) = n - m^*$.

5. **Extra Credit:** Let E^* be an edge cover of minimum size.

   (a) How many connected components are there in $G = (V, E^*)$ in terms of $|E^*|$ and n? Justify your answer.

   Answer: Note that E^* is acyclic because if there is a cycle we can remove at least one edge and still have an edge cover. Hence, it there are r connected components the number of edges will be $n - r = |E^*|$ and therefore $r = n - |E^*|$.

   (b) Using your answer for part (5a), prove that your algorithm for part (4) finds the smallest edge cover.

   Answer: m^* is at least $r = n - |E^*|$ since we can pick an edge from each connected component in the matching. Hence $m^* \geq n - |E^*|$ and so $|E^*| \geq n - m^*$. Therefore the edge cover found in part (4) is best possible.